あいまいな位置情報に基づく最近傍問合せの処理手法

飯島　裕一[†*]　　石川　佳治[†,††a)]

Processing Methods for Nearest Neighbor Queries Based on Imprecise Location Information

Yuichi IIJIMA[†*] and Yoshiharu ISHIKAWA[†,††a)]

あらまし　移動ロボットやモバイルセンサネットワークなどの位置情報を利用したアプリケーションでは，最近傍問合せは重要な問合せとして位置づけられている．しかし，制御ノイズや測定誤差などの理由により問合せを行うオブジェクトの位置があいまいな位置情報としてしか得ることができない場合が存在する．そのため，位置のあいまい性を考慮した問合せ処理手法が必要とされている．そこで本論文では，問合せを行うオブジェクトの位置が正規分布の確率密度関数によってあいまいな位置情報として表現されている状況における最近傍問合せの処理手法を提案する．まず通常の最近傍問合せを拡張した確率的最近傍問合せを定義し，この問合せを効率的に処理するための戦略として二つの問合せ戦略を提案する．どちらの戦略も，問合せの対象となるオブジェクトの集合から明らかに問合せを満たさないといえるものを求めることで計算コストの削減を図るが，その方法が異なっている．これら二つの戦略にそれらのハイブリッド方式の戦略を加えた三つの戦略に対して，実験によって性能の比較を行った結果についても報告する．

キーワード　空間データベース，最近傍問合せ，あいまいな位置，正規分布

1. まえがき

近年，移動ロボットやモバイルセンサネットワークなどの分野において，あいまいな位置情報に基づく空間問合せの処理技術に対する必要性が高まってきている．移動ロボットは通常，センサ信号や移動履歴などをもとに統計的な手法を用いて自身の位置を継続的に推定する [1] が，センサの測定誤差やモータの制御ノイズなどのために正確な位置の推定は容易ではなく，誤差を伴った推定となる．また，モバイルセンサネットワークにおいては GPS を用いて各センサの位置情報を得ることが一般的であるが，電波状況によっては必ずしも十分な測位精度が得られるとは限らない [2]．その上，GPS による位置情報の取得は多くの電力を消費するため，各センサが電池で駆動しているような場合には極力避けたいという要求もある．以上のように，現実世界のオブジェクトの位置はあいまいな位置情報としてしか得ることができない場合が多いため，位置のあいまいさを考慮した問合せ処理手法が必要とされており，その研究が盛んになってきている．

本研究では，あいまいな位置をもつオブジェクトが，自らの位置から最も近くにあるオブジェクトを検索するために最近傍問合せを行うという状況を対象とする．具体的には，問合せを行うオブジェクト（以降，問合せオブジェクト）の位置が正規分布で表現され，問合せの対象となるオブジェクト（以降，データオブジェクト）が確定的な位置で表される点データである状況を扱う．ただし，各データオブジェクトは通信能力や計算能力を有していないものとする．対象とする問合せとして，ユークリッド距離に基づく通常の最近傍問合せを拡張した確率的最近傍問合せ (probabilistic nearest neighbor query，PNNQ) を定義し，この問合せを効率的に処理するために二つの問合せ戦略を提案する．実験では，二つの戦略にそれらのハイブリッド戦略を加えた三つの戦略について，様々なパラメー

† 名古屋大学大学院情報科学研究科，名古屋市
Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, 464–8601 Japan
†† 名古屋大学情報基盤センター，名古屋市
Information Technology Center, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, 464–8601 Japan
* 現在，アイホン株式会社
a) E-mail: ishikawa@itc.nagoya-u.ac.jp

タ設定のもとで各戦略の比較を行う．

本論文ではまず，**2.** で関連研究を紹介する．次に，**3.** で確率的最近傍問合せを定義し，続く **4.** でその処理手法を提案する．**5.** では実験に基づいて各戦略の性能を比較する．最後に **6.** でまとめを行う．

2. 関連研究

近年，位置のあいまいさを考慮した問合せ処理に注目が集まっており，多くの研究がなされている．対象とされる問合せの種類は様々であるが，中心になっているのは範囲問合せ [2]〜[6] と最近傍問合せ [3], [7]〜[9] である．これらの研究において，あいまいさを表現するためのモデルはそれぞれ異なっており，例えば, [2] では最も単純に，あいまいなオブジェクトの位置が一様分布に従い，更にその存在範囲が与えられることを前提としている．その一方で, [4], [5], [9] などのように任意の確率分布の使用を認めて一般性を高めている研究もある．

これらの研究では通常，あいまいなオブジェクトに対して，そのオブジェクトが内部に指定値以上の確率で存在することが保証されている領域が与えられることが前提となる．この領域は *uncertainty region* と呼ばれ，本研究でも一部この概念を導入している．ただし，本研究では位置のあいまいさが任意の分布ではなく正規分布に基づいている状況を対象とする．

正規分布は統計やパターン認識 [10] などの分野で非常によく用いられる確率密度関数である．移動オブジェクトデータベースの分野では，移動オブジェクトの位置のあいまいさを正規分布によって表現するアイデアが [2] で提案されている．また，移動ロボットの分野では，センサや移動履歴などをもとにしたローカライゼーション（自己位置推定）にしばしば正規分布が使用される．特に，正規分布に基づく確率過程を前提としたカルマンフィルタは，移動ロボットのローカライゼーションにおける伝統的なアプローチである [1]．カルマンフィルタによるロボットの位置推定では，正規分布で表現された時刻 t における位置を入力とし，正規分布でモデル化された雑音を含むセンサデータを用いて時刻 $t+1$ における位置を推定する．出力される推定結果もまた正規分布で表現されており，これを再度入力として与えることで，再帰的に推定を行う．カルマンフィルタに基づく位置推定は移動ロボットの分野以外でも活用されている [11], [12].

GPS を用いた位置情報の取得と利用も今日では一般化しているが，測位誤差が正規分布に従わないため本研究の想定とは合致しない．ただし，GPS による測位結果を正規分布として大まかにモデル化する研究もある [2], [12].

移動ロボットなどの応用を考えると，位置が正規分布に従うという想定には一般性があり，対象領域によっては十分な妥当性があるといえる．したがって，本研究では正規分布に特化した処理技術に焦点を合わせる．任意の確率分布の使用を認めた研究に比べれば汎用性は若干劣ることになるが，前述のとおり正規分布は位置のあいまいさの表現によく用いられる分布であり，汎用性は高いと考えられる．むしろ，正規分布の性質を効果的に用いた問合せ処理アルゴリズムの開発が可能となるという利点が大きい．

あいまいさを考慮した空間問合せ処理に関する研究は各々の対象とする状況から以下の 3 種類に分類できる．

- データオブジェクトのみあいまい [2], [3], [5], [8]
- 問合せオブジェクトのみあいまい [6]
- 両オブジェクトともにあいまい [4], [7], [9]

[3] は uncertainty region を用いて候補の絞り込みを行った後，数値積分により問合せの条件を満たす確率を計算するというアプローチに基づいており，本研究でも一部同様のアプローチをとる．[8] は一次元のあいまいなオブジェクトに対する最近傍問合せを検討しており，そこで提案されている問合せは確率のしきい値が与えられるという点で本研究と関係している．この問合せの特徴は，正確な確率値の計算を行わなくても確率の範囲の計算だけで候補を絞り込むことができる点にある．これらの研究はデータオブジェクトのみがあいまいであるという状況を対象としているが，サンプリング手法を用いたアプローチ [7] や k-最近傍問合せの処理手法 [9] など，問合せオブジェクトの方もあいまいである状況を対象とした最近傍問合せの処理手法もいくつか提案されている．

一方，本研究が対象とするのは問合せオブジェクトのみがあいまいであるという状況である．[6] は本研究グループによる研究であり，本研究が対象とする状況と同様の状況における範囲問合せの処理手法を提案している．本研究ではそのアイデアを一部導入しているが，対象とする問合せが最近傍問合せであるため，その特徴を考慮した改良や新しい技術が必要となる．最近傍問合せの処理にはデータオブジェクトに対するボロノイ図 [13] を用いるのが一般的であり，本研究でもボロノイ図を効果的に使用することで効率的な問合せ

処理を実現する．データオブジェクトがあいまいである状況に対してボロノイ図を適用する場合には適切な拡張が必要となるが，現状ではそのような研究はまだなされていない．しかしながら本研究では，データオブジェクトの位置が確定的である状況を扱うため通常のボロノイ図を利用できる．[9] においても，データオブジェクトの位置が確定的である状況に限ってボロノイ図が用いられている．

3. 確率的最近傍問合せの定義

本研究では問合せオブジェクトの位置が正規分布の確率密度関数によってあいまいな位置情報で表現されている状況を対象とするため，あるオブジェクトが最近傍オブジェクトであるかどうかは確率的に決まる．このことを，一次元の場合を例に以下で説明する．

［例 1］ あいまいな位置情報に基づく最近傍問合せ

一次元直線上に問合せオブジェクト q とデータオブジェクト a，b，c，d，e が存在している状況を考える．a，b，c，d，e は図 1 に示すような位置に固定されており，その位置にあいまい性はないとする．一方，q の位置は不定であり，その x 座標 x_q の確率密度関数 $p_q(x)$ が平均 0，分散 1 の正規分布で表されているとする．図中の曲線はその $p_q(x)$ を示している．このとき，q が自身の位置から最も近いオブジェクトを求めるために最近傍問合せを発行したとする．仮に，q の位置があいまいではなく，$x = 0$ に固定されているとすれば，問合せ結果は $\{c\}$ ということになる．しかしながら，ここでは q の位置が正規分布に従うという状況を想定しているため，q は直線上のどこにでも位置する可能性があり，問合せ結果が一意に決まらなくなる．例えば，q の位置が $x = -1$ にあるとすれば最近傍オブジェクトは b となるが，$x = 1$ にあるとすれば最近傍オブジェクトは d となる．最近傍問合せの問合せ結果は問合せオブジェクトとデータオブジェクトの位置関係によって決まるため，問合せオブジェクトの位置が確率的である場合，問合せ結果も確率的になる．

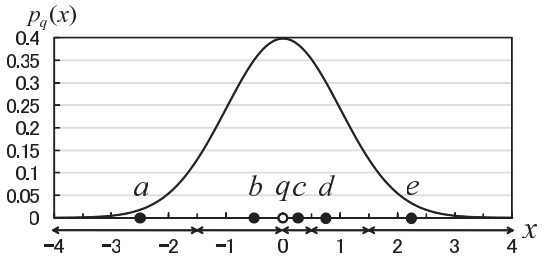

図 1 あいまいな位置情報に基づく最近傍問合せ
Fig. 1 Nearest neighbor query based on imprecise location information.

このような状況に対応するためには，各データオブジェクトが最近傍オブジェクトになる確率を考慮して問合せ結果が定まるような最近傍問合せを定義する必要がある．ユークリッド距離に基づく通常の最近傍問合せを拡張する形で，以下にこれを定義する．

［定義 1］ 確率的最近傍問合せ

d 次元空間において，問合せオブジェクト q の位置が d 次元ベクトルの座標値 $\boldsymbol{x}$ をもつ確率が，d 次元正規分布の確率密度関数により，

$$p_q(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{q})^t \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{q})\right] \tag{1}$$

で表現されるとする．ただし，$\boldsymbol{\Sigma}$ は $d \times d$ の共分散行列，$\boldsymbol{q}$ は分布の平均，t はベクトルの転置を表す．このような q が与えられたとき，q とのユークリッド距離がすべてのデータオブジェクトのうちで最小となる（q の最近傍オブジェクトとなる）確率が θ $(0 < \theta < 1)$ 以上であるオブジェクトの集合を返す問合せを確率的最近傍問合せ $PNNQ(q, \theta)$ と定義する．データオブジェクトの集合を $\mathcal{O}$ とするとき，$o \in \mathcal{O}$ が q の最近傍オブジェクトとなる確率 $\Pr_{NN}(q, o)$ は以下の式で表される．

$$\begin{aligned}&\Pr_{NN}(q, o)\\ &\quad = \Pr(\forall o' \in \mathcal{O}, o' \neq o, \|\boldsymbol{x}-\boldsymbol{o}\|^2 \leq \|\boldsymbol{x}-\boldsymbol{o}'\|^2)\end{aligned} \tag{2}$$

これを用いて $PNNQ(q, \theta)$ は以下の式で表現できる．

$$PNNQ(q, \theta) = \{n \mid n \in \mathcal{O}, \Pr_{NN}(q, n) \geq \theta\} \tag{3}$$

問合せの入力として与えられるのは，問合せオブジェクト q の情報と確率のしきい値 θ である．q の情報とは，具体的には q の位置が従う正規分布の確率密度関数 $p_q(\boldsymbol{x})$ のことである．更にいえば，式 (1) に示すとおり，$p_q(\boldsymbol{x})$ は共分散行列 $\boldsymbol{\Sigma}$ と平均 $\boldsymbol{q}$ により一意に決まる．したがって，$\boldsymbol{\Sigma}$，$\boldsymbol{q}$，θ の三つのパラメータが問合せ時に与えられることになる．

問合せ結果に含まれるオブジェクト（以下，解オブジェクト）の個数は θ の設定値に依存する．具体的には，θ を高く設定するほど減少し，低く設定するほど増加する．ただし，θ の設定が高すぎた場合には空の

結果が返され，低すぎた場合には多数の解オブジェクトが返されることになる．

以下に，確率的最近傍問合せの例を示す．

［例 2］ 一次元の確率的最近傍問合せ

例 1 において，q が $PNNQ(q, 0.2)$ を発行したとする．これは，$\Pr_{NN}(\cdot)$ が 0.2 以上であるデータオブジェクトを求める問合せである．ここで，a が q の最近傍オブジェクトになるのは，q が $x \leq -1.5$ の範囲に位置している場合である．その確率は x_q の確率密度関数をこの範囲で積分することで得られるため，$\Pr_{NN}(q, a) = \int_{-\infty}^{-1.5} p_q(x)dx = 0.067$ となる．同様に，b が q の最近傍オブジェクトとなるのは $-1.5 \leq x_q \leq 0$ の場合であり，その確率は $\Pr_{NN}(q, b) = \int_{-1.5}^{0} p_q(x)dx = 0.433$ である．c, d, e についてもそれぞれ計算すると，$\Pr_{NN}(q, c) = \int_{0}^{0.5} p_q(x)dx = 0.191$，$\Pr_{NN}(q, d) = \int_{0.5}^{1.5} p_q(x)dx = 0.242$，$\Pr_{NN}(q, e) = \int_{1.5}^{\infty} p_q(x)dx = 0.067$ となる．以上より，問合せ結果は $\{b, d\}$ と求まる．

上記の例の場合では，正規分布の平均 0 に最も近い c が問合せ結果には含まれず，c よりも 0 から遠い b, d が問合せ結果に含まれるという結果となった．これは，各オブジェクトの「q の最近傍オブジェクトになる範囲」（以下，勢力範囲）の大きさが，b は 1.5，d は 1 であるのに対して，c は 0.5 と小さいことが影響している．各オブジェクトの勢力範囲を図 1 の x 軸の下に矢印で示した．a，e は b，d よりも勢力範囲が大きいが，例の問合せ結果には含まれない．これは，b, d に比べて a，e が正規分布の平均 0 から遠くに位置していることが影響している．確率的最近傍問合せにおいて，あるオブジェクトが解オブジェクトになるかどうかは，そのオブジェクトの勢力範囲の大きさと正規分布の平均からの距離に大きく依存する．

これまで一次元の場合を例に説明してきたが，本研究の提案手法は一般に二次元以上の場合を対象としていることを断わっておく．ここでは例示のため一次元の場合を説明したが，一次元の場合の問合せ処理は容易であるため，二次元以上の場合にも適用可能なアルゴリズムを示す．二次元以上の場合について，各オブジェクトの勢力範囲を示した図が，**2.** で触れたボロノイ図である．次章で提案手法について説明する．

4. 提案手法

4.1 基本的なアイデア

本手法ではボロノイ図と呼ばれる，空間中の複数の点に対して，どの点に一番近いかによって空間を分割した図を用いる．例として図 2 に点 a〜j に対するボロノイ図を示す．各点の勢力範囲はボロノイ領域と呼ばれ，図 2 の陰影部分は g のボロノイ領域 V_g を示している．ボロノイ図の定義から，データオブジェクト o のボロノイ領域 V_o 内に q が位置する場合に o は q の最近傍オブジェクトとなる．したがって，o が q の最近傍オブジェクトとなる確率は q が V_o 内に位置する確率と言い換えられる．つまり，$\Pr_{NN}(q, o)$ は式 (1) に示した $p_q(\boldsymbol{x})$ を領域 V_o で積分することで計算できる．以上の事実を踏まえ，式 (2) に示した $\Pr_{NN}(q, o)$ の計算式は以下のように書き換えられる．

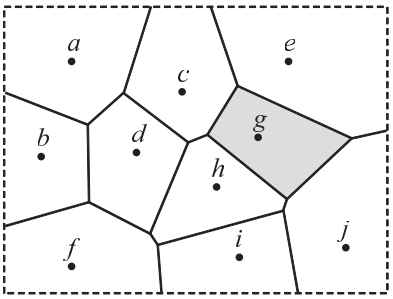

図 2 ボロノイ図
Fig. 2 Voronoi diagram.

$$\Pr_{NN}(q, o) = \int_{\boldsymbol{x} \in V_o} p_q(\boldsymbol{x})d\boldsymbol{x} \tag{4}$$

$p_q(\boldsymbol{x})$ として本研究では正規分布を対象としているが，その積分は解析的には計算できないため，コストの高い数値積分が必要となる．その上，式 (4) の計算の場合，各ボロノイ領域の形状が複雑な多面体であることも計算コストを高める要因となる．したがって，すべてのデータオブジェクトに対して直接的に $\Pr_{NN}(\cdot)$ を求めることは現実的ではない．そこで本手法では，明らかに $\Pr_{NN}(\cdot)$ が θ に満たないといえるオブジェクトを除去（フィルタリング）し，残った候補オブジェクトに対してのみ $\Pr_{NN}(\cdot)$ を計算する．このアイデアに基づく問合せ戦略を 2 種類提案する．

4.2 問合せ戦略 1

本戦略では **2.** で紹介した uncertainty region [5] の概念を活用する．具体的には以下で定義する，θ-領域 [6] という領域を用いてフィルタリングを行う．

［定義 2］ θ-領域

$(\boldsymbol{x} - \boldsymbol{q})^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{q}) \leq r^2$ を満たす楕円体領域での $p_q(\boldsymbol{x})$ の積分を考える．与えられた θ $(0 < \theta < 1/2)$

に対し，積分値が $1-2\theta$ になるような r の値を r_θ とする．式で表現すると以下のとおりである．

$$\int_{(\boldsymbol{x}-\boldsymbol{q})^t\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{q})\le r_\theta^2} p_q(\boldsymbol{x})d\boldsymbol{x} = 1-2\theta \qquad (5)$$

r_θ により以下の式で定まる楕円体領域を θ-領域と呼ぶ．

$$(\boldsymbol{x}-\boldsymbol{q})^t\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{q}) \le r_\theta^2 \qquad (6)$$

θ-領域は問合せ時に与えられるパラメータに依存するため，その導出は問合せ時に動的に行う必要がある．単純な方法として，問合せ時に様々な r の値に対して対応する楕円体領域での $p_q(\boldsymbol{x})$ の積分値を数値積分によって計算し，その値が $1-2\theta$ となるような $r=r_\theta$ を見つけるという方法が考えられるが，計算コストの面で現実的ではない．そこで，楕円体領域での積分を球領域での積分に変換する．まず，式 (1) において $\boldsymbol{q}=\boldsymbol{0}$，$\boldsymbol{\Sigma}=\mathbf{I}$ とした，標準正規分布の確率密度関数

$$p_{\mathrm{norm}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}}\exp\left[-\frac{1}{2}\|\boldsymbol{x}\|^2\right] \qquad (7)$$

を考える．これを用いて以下の性質を導出することができる．証明は [6] を参照されたい．

［性質 1］ 原点を中心とした半径 r の球領域 $\|\boldsymbol{x}\|^2 \le r^2$ での $p_{\mathrm{norm}}(\boldsymbol{x})$ の積分を考える．与えられた θ に対し，積分値が $1-2\theta$ になるような半径を $\tilde{r}_\theta$ と定義する．式で表現すると以下のとおりである．

$$\int_{\|\boldsymbol{x}\|^2\le\tilde{r}_\theta^2} p_{\mathrm{norm}}(\boldsymbol{x})d\boldsymbol{x} = 1-2\theta \qquad (8)$$

このとき，与えられた θ に対して以下の式が成り立つ．

$$r_\theta = \tilde{r}_\theta \qquad (9)$$

この性質は，与えられた θ に対して，式 (8) に基づいて $\tilde{r}_\theta$ を計算すれば，その値がそのまま θ-領域を定める r_θ になっているということを示している．しかしながら，$p_{\mathrm{norm}}(\boldsymbol{x})$ の積分値は解析的に求めることができないため，θ から直接 r_θ を計算することはできない．そこで逆に，適当な半径の値を選んでその半径をもつ球領域での $p_{\mathrm{norm}}(\boldsymbol{x})$ の積分値を数値積分によって計算するということを，様々な半径の値に対して行うことで，積分値から得られる θ とそのときの半径 r_θ の対応表を事前に作成しておくことにする．この表を引くことで与えられた θ に対応する r_θ を素早く得ることが可能となる．このようなアイデアは [5] でも導入されており，表は *U-catalog* と呼ばれている．

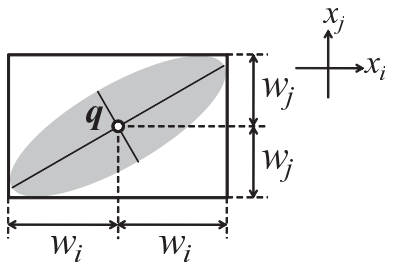

図 3 包囲方形の利用
Fig. 3 Using bounding box.

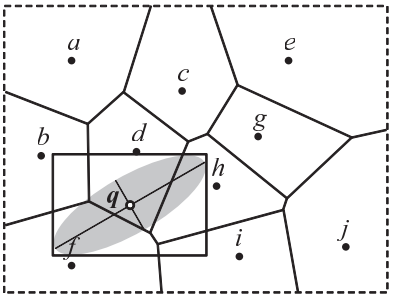

図 4 包囲方形とボロノイ領域
Fig. 4 Bounding box and Voronoi regions.

楕円体の形状をもつ θ-領域を直接フィルタリングに利用することは難しいため，図 3 に示すように，座標軸に平行な θ-領域に外接する方形を用いることにする．この包囲方形は，分布の平均 $\boldsymbol{q}$ から i 番目の次元について大小方向に w_i の幅をもつものとする．w_i に関して以下の性質が成り立つ．証明は [6] を参照されたい．

［性質 2］ w_i $(i=1,2,\ldots,d)$ の値は

$$w_i = \sigma_i r_\theta \qquad (10)$$

として与えられる．ただし，σ_i は i 番目の次元に関する標準偏差に相当し，$(\boldsymbol{\Sigma})_{ii}$ を共分散行列 $\boldsymbol{\Sigma}$ の i 行 i 列の値としたとき以下の式で定義される．

$$\sigma_i = \sqrt{(\boldsymbol{\Sigma})_{ii}} \qquad (11)$$

図 4 を用いて本戦略のアイデアを説明する．図の陰影部分が θ-領域であり，外接する方形によって包囲されている．このとき，ボロノイ領域が包囲方形と重なりをもたないオブジェクト，すなわち a，c，e，g，j を解の候補から除去することができる．理由は以下のとおりである．まず，θ-領域の定義から，包囲方形の外

アルゴリズム 1 戦略 1 に基づく確率的最近傍問合せ

```
1: procedure PNNQ-1(q, Σ, θ)
2:     C ← ∅, sum ← 0
3:     r_θ ← lookup(θ)
4:     {σ_i}_{i=1}^d 及び r_θ から図 3 に示した包囲方形を導出
5:     ボロノイ領域が包囲方形と重なりをもつオブジェクトを
       検索して C に挿入
6:     foreach o ∈ C do
7:         Pr_NN(q, o) ← ∫_{x∈V_o} p_q(x)dx    ▷ 数値積分による
8:         sum ← sum + Pr_NN(q, o)
9:         if Pr_NN(q, o) ≥ θ then
10:            output o
11:        end if
12:        if sum > 1 − θ then
13:            return
14:        end if
15:    end for
16: end procedure
```

側の領域全体での $p_q(\boldsymbol{x})$ の積分値は $1-(1-2\theta)=2\theta$ 未満である．また，$p_q(\boldsymbol{x})$ は分布の平均 $\boldsymbol{q}$ について点対称な分布であるため，ボロノイ領域 V_o と $\boldsymbol{q}$ について対称な領域 V_o' での $p_q(\boldsymbol{x})$ の積分値は V_o での積分値に等しい．これらの事実により，包囲方形と重なりをもたないボロノイ領域での $p_q(\boldsymbol{x})$ の積分値は 2 倍しても 2θ 未満ということになる．すなわち，ボロノイ領域が包囲方形と重なりをもたないオブジェクトは $\mathrm{Pr}_{NN}(\cdot)$ が θ 以上になることはないとして除去できる．

本戦略のアルゴリズムをアルゴリズム 1 に示す．ボロノイ領域が包囲方形と重なりをもつオブジェクトを検索して候補オブジェクトとした後，すべての候補オブジェクトに対して，数値積分により $\mathrm{Pr}_{NN}(\cdot)$ を求め，θ 以上であれば出力するという流れで処理を行う．ただし，U-catalog の作成と各ボロノイ領域の頂点の座標のファイルへの記録を事前に行っておくものとする．

3 行目の関数 lookup は U-catalog を引いて適切な r_θ を返す関数である．注意として，与えられた θ に一致するエントリが U-catalog 中に存在しない場合には，$\theta^* < \theta$ を満たす最大の θ^* をもつエントリの r_θ の値 r_θ^* を返す．これにより，多少余分に候補オブジェクトが検索されることになるが，結果の正しさは保証される．5 行目でボロノイ領域が包囲方形と重なりをもつオブジェクトを検索する必要があるが，各ボロノイ領域は複雑な形状をとるため，そのようなオブジェクトだけを正確に検索する処理はコストが高い．解決策としては，各ボロノイ領域に対して座標軸に平行な包囲方形を求め，この包囲方形が θ-領域の包囲方形と重なりをもつオブジェクトを候補オブジェクトとすることが考えられる．これにより，若干候補オブジェクトの個数が増加することになるが，フィルタリングに要する時間を減らすことができる．予備実験の結果，候補オブジェクトの増加による確率計算の処理時間の増加分をフィルタリングの処理時間の減少分が上回る見込みが濃厚であったため，**5.** の実験ではこの方法に基づいて実装を行った．12 行目の条件が満たされると，その時点で残っている候補については $\mathrm{Pr}_{NN}(\cdot)$ が θ 以上である可能性がなくなるため，処理を終了できる．

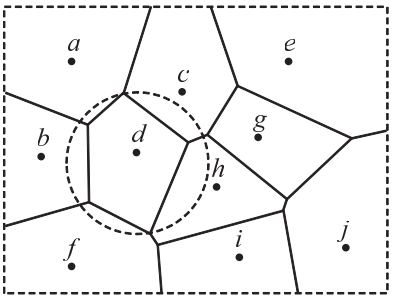

図 5　最小包含球
Fig. 5　Smallest enclosing sphere.

4.3　問合せ戦略 2

本戦略では，各データオブジェクトに対して $\mathrm{Pr}_{NN}(\cdot)$ の上限値を求めることによりフィルタリングを行う．上限値の計算はボロノイ領域の最小包含球 (smallest enclosing sphere，SES) を利用して行う．例としてボロノイ領域 V_d の最小包含球を図 5 に示す．最小包含球の領域で $p_q(\boldsymbol{x})$ を積分すると，その値は $\mathrm{Pr}_{NN}(\cdot)$ の上限値とみなすことができる．球領域での積分値は事前に表を作成しておくことで簡単に求められるため，最小包含球による上限値の計算は高速なフィルタリング処理の実現に有効である．はじめに式 (1) の共分散行列 $\boldsymbol{\Sigma}$ が単位行列であるという単純な場合を考え，次にアイデアを一般の場合に拡張する．

4.3.1　$\boldsymbol{\Sigma}=\mathbf{I}$ の場合

本項では $\boldsymbol{\Sigma}$ が単位行列である場合について考える．この場合の $p_q(\boldsymbol{x})$ は，$p_{\mathrm{norm}}(\boldsymbol{x})$ を $\boldsymbol{q}$ が中心となるように平行移動したものに等しい．

最小包含球の半径や中心の座標はオブジェクトごとに様々であるため，異なる最小包含球に対して，その領域での $p_q(\boldsymbol{x})$ の積分値を素早く導出できるように表を事前に作成しておく．表の作成にあたり，図 6 に示すような，原点から距離 α の点を中心とする半径 δ の d 次元の球 R を考える．このとき，$p_{\mathrm{norm}}(\boldsymbol{x})$ を R の

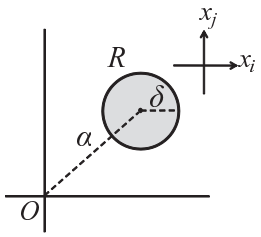

図 6　球 R
Fig. 6　Sphere R.

α	δ	$\pi(\alpha,\delta)$
0.0	0.1	$\cdots$
0.0	0.2	$\cdots$
$\vdots$	$\vdots$	$\vdots$
1.0	0.1	$\cdots$
1.0	0.2	$\cdots$
$\vdots$	$\vdots$	$\vdots$

図 7　戦略 2 の U-catalog
Fig. 7　U-catalog for strategy 2.

領域で積分した値を以下の式で表す．

$$\pi(\alpha,\delta)=\int_{\boldsymbol{x}\in R} p_{\mathrm{norm}}(\boldsymbol{x})d\boldsymbol{x} \tag{12}$$

$p_{\mathrm{norm}}(\boldsymbol{x})$ の等確率面は球形であり，原点からの距離と半径がともに等しい任意の球領域での積分値は一定であるため，このような表記を用いることができる．異なる α と δ の値の組合せに対して，数値積分により $\pi(\alpha,\delta)$ を計算し，図 7 のような表に結果を格納する．この表も戦略 1 で用いた表と同様に U-catalog と呼ぶ．U-catalog は (α,δ) のペアを与えると対応する積分値を返す．

次に，U-catalog の使用方法を説明する．$\mathrm{Pr}_{NN}(q,o)$ の上限値を求めるためには，ボロノイ領域 V_o の最小包含球を SES_o として $\int_{\boldsymbol{x}\in SES_o} p_q(\boldsymbol{x})d\boldsymbol{x}$ の値を計算すればよい．この値は，$\boldsymbol{q}$ から SES_o の中心までの距離を α_o，SES_o の半径を δ_o としたときの $\pi(\alpha_o,\delta_o)$ に等しいため，(α_o,δ_o) に一致するエントリを U-catalog から検索すれば簡単に得られる．得られた値が θ 以下である場合には o を棄却できる．一方，この値はあくまでも $\mathrm{Pr}_{NN}(q,o)$ の上限値であるため，値が θ より大きいからといって o が解になるとは限らない．そのため，U-catalog を引いて得られた値が θ より大きいオブジェクトは候補オブジェクトとして残す．

U-catalog のエントリ数は有限であるため，戦略 1 の場合と同様に，与えられた (α_o,δ_o) に一致するエントリが存在しないことがある．このような場合には，$\alpha_o^*\le\alpha_o$ かつ $\delta_o^*\ge\delta_o$ を満たすような (α_o^*,δ_o^*) をもつエントリのうちで，$\pi(\alpha,\delta)$ の値が最小のものを見つける．つまり，(α_o,δ_o) に対応する積分値よりは大きいが，できる限りそれに近い値を返すようなエントリを見つける．これにより，多少候補オブジェクトとして残る可能性が高まるが，結果の正しさは保証される．次に，これまで説明したアイデアを一般化する．

4.3.2　一般の場合

本項では $\boldsymbol{\Sigma}$ が任意である場合について考える．この場合の $p_q(\boldsymbol{x})$ の等確率面は楕円体の形状をとるため，先の場合のように単純に (α_o,δ_o) のペアによって任意の球領域での積分値を表現することは不可能であり，表を用いて積分値を求めるわけにはいかない．そこで，$p_q(\boldsymbol{x})$ の上限の関数 $p_q^\top(\boldsymbol{x})$ を導入する．

［定義 3］　上限の関数

共分散行列の逆行列 $\boldsymbol{\Sigma}^{-1}$ のスペクトル分解を

$$\boldsymbol{\Sigma}^{-1}=\sum_{i=1}^{d}\lambda_i\boldsymbol{v}_i\boldsymbol{v}_i^t \tag{13}$$

と表す．ただし，λ_i と $\boldsymbol{v}_i$ はそれぞれ i 番目の固有値と固有ベクトルである．このとき，

$$\lambda^\top=\min\{\lambda_i\} \tag{14}$$

と定義する．共分散行列の固有値はすべて 0 より大きいため，$\lambda^\top>0$ が成り立つことに注意する．ここで，

$$\mathbf{M}^\top=\lambda^\top\sum_{i=1}^{d}\boldsymbol{v}_i\boldsymbol{v}_i^t=\lambda^\top\mathbf{I} \tag{15}$$

と定義したとき，式 (1) の $\boldsymbol{\Sigma}^{-1}$ を行列 $\mathbf{M}^\top$ で置き換えることで得られる関数を $p_q^\top(\boldsymbol{x})$ と定義する．

$$p_q^\top(\boldsymbol{x})=\frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\left[-\frac{\lambda^\top}{2}\|\boldsymbol{x}-\boldsymbol{q}\|^2\right] \tag{16}$$

$p_q^\top(\boldsymbol{x})$ の等確率面は球形である．ただし，空間全体での積分値が 1 とはならないため，厳密には $p_q^\top(\boldsymbol{x})$ は確率密度関数ではない．$p_q^\top(\boldsymbol{x})$ は以下の性質をもつ．

［性質 3］　任意の $\boldsymbol{x}$ に対して，以下の式が成り立つ．

$$p_q(\boldsymbol{x})\le p_q^\top(\boldsymbol{x}) \tag{17}$$

この性質を満たし，等確率面が球形の関数のうちで最良のものが $p_q^\top(\boldsymbol{x})$ である．つまり，$p_q^\top(\boldsymbol{x})$ は $p_q(\boldsymbol{x})$ の上限を与えている．図 8 に同じ確率に対する $p_q(\boldsymbol{x})$

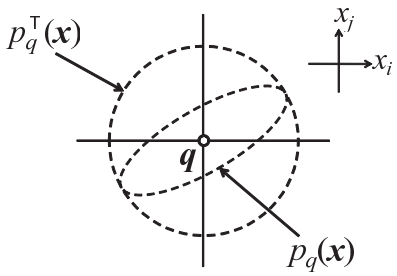

図 8　$p_q(\boldsymbol{x})$ と $p_q^\top(\boldsymbol{x})$
Fig. 8　$p_q(\boldsymbol{x})$ and $p_q^\top(\boldsymbol{x})$.

アルゴリズム 2 戦略 2 に基づく確率的最近傍問合せ

```
1: procedure PNNQ-2(q, Σ, θ)
2:     C ← ∅, sum ← 0
3:     λ^⊤ 及び |Σ| を Σ から計算
4:     foreach o ∈ O do
5:         q から SES_o の中心までの距離 α_o を計算
6:         π(α_o√λ^⊤, δ_o√λ^⊤) ← lookup(α_o√λ^⊤, δ_o√λ^⊤)
7:         IV_SES_o ← π(α_o√λ^⊤, δ_o√λ^⊤)/(λ^⊤)^{d/2}|Σ|^{1/2}
8:                                   ▷ SES_o での p_q^⊤(x) の積分値
9:         if IV_SES_o > θ then
10:            C ← C ∪ {o}
11:        end if
12:    end for
13:    C 中のオブジェクトを IV_SES_o の降順でソート
14:    foreach o ∈ C do                          ▷ 先頭から順に
15:        Pr_NN(q, o) ← ∫_{x∈V_o} p_q(x)dx      ▷ 数値積分による
16:        sum ← sum + Pr_NN(q, o)
17:        if Pr_NN(q, o) ≥ θ then
18:            output o
19:        end if
20:        if sum > 1 − θ then
21:            return
22:        end if
23:    end for
24: end procedure
```

と $p_q^\top(\boldsymbol{x})$ の等確率面を示す．$p_q^\top(\boldsymbol{x})$ については等確率面が球形であるため，表を用いて任意の球領域での積分値を求めることができる．その上，表は **4.3.1** で説明した U-catalog をそのまま使用することができる．具体的には，$(\alpha_o\sqrt{\lambda^\top}, \delta_o\sqrt{\lambda^\top})$ に一致するエントリを U-catalog から検索し，得られた $\pi(\alpha_o\sqrt{\lambda^\top}, \delta_o\sqrt{\lambda^\top})$ を $(\lambda^\top)^{d/2}|\boldsymbol{\Sigma}|^{1/2}$ で割ることで求められる．証明は [6] を参照されたい．性質 3 より，同じ領域で積分した場合に，$p_q^\top(\boldsymbol{x})$ の積分値が $p_q(\boldsymbol{x})$ のそれを下回ることはないため，最小包含球領域での $p_q^\top(\boldsymbol{x})$ の積分値が θ 以下であるオブジェクトは $\mathrm{Pr}_{NN}(\cdot)$ が θ 以上になることはないとして棄却できる．

本戦略のアルゴリズムをアルゴリズム 2 に示す．ただし，U-catalog の作成と各ボロノイ領域の頂点の座標，最小包含球の中心点及び半径のファイルへの記録を事前に行っておくものとする．6 行目の関数 lookup は，$(\alpha_o\sqrt{\lambda^\top}, \delta_o\sqrt{\lambda^\top})$ に一致するエントリを U-catalog から検索して $\pi(\alpha_o\sqrt{\lambda^\top}, \delta_o\sqrt{\lambda^\top})$ を返す関数である．一致するエントリが U-catalog 中に存在しない場合には，**4.3.1** で説明したとおり，$(\alpha_o\sqrt{\lambda^\top}, \delta_o\sqrt{\lambda^\top})$ に対応する積分値よりは大きいが，できる限りそれに近い値を返す．13 行目のソートにより，最小包含球領域での $p_q^\top(\boldsymbol{x})$ の積分値が大きいオブジェクトから順に $\mathrm{Pr}_{NN}(\cdot)$ を計算できる．この値が大きいオブジェクトはボロノイ領域での $p_q(\boldsymbol{x})$ の積分値，すなわち $\mathrm{Pr}_{NN}(\cdot)$ も大きいと考えられるため，順序を考慮しない場合よりも早く処理を終了できる可能性が高い．

5. 実　　験

5.1 実験方法

使用したデータは米国加州ロングビーチの道路の線分データ [14] から各線分の中点を抽出して作成したデータである．データ数は 50,501 で，各点は $[0, 1000]^2$ の二次元空間上に位置するように正規化されている．各点をデータオブジェクトとして，二つの問合せ戦略にそれらのハイブリッド戦略を加えた三つの戦略を対象に，$PNNQ(q, \theta)$ に対する性能評価を行った．

ハイブリッド戦略は，戦略 1 のフィルタリングと戦略 2 のフィルタリングを組み合わせたフィルタリングを行う戦略である．具体的には，はじめに戦略 1 のフィルタリングを行った後，残ったオブジェクトに対して戦略 2 のフィルタリングを行う．そのため，得られる候補オブジェクトの集合は，戦略 1 のフィルタリングによって得られる候補オブジェクトの集合と戦略 2 のフィルタリングによって得られる候補オブジェクトの集合の積集合になる．戦略 1 のフィルタリングを行ってから戦略 2 のフィルタリングを行うという順序には理由があるが，これについては実験の結果を踏まえながら **5.2.1** で述べる．

式 (1) の共分散行列 $\boldsymbol{\Sigma}$ の設定は以下を標準とした．

$$\boldsymbol{\Sigma} = \gamma \begin{bmatrix} 7 & 2\sqrt{3} \\ 2\sqrt{3} & 3 \end{bmatrix}$$

これにより，$p_q(\boldsymbol{x})$ の等確率線の形状は長軸と短軸の比が 3 : 1 で傾き 30° の楕円となる．係数 γ は分布のあいまいさの程度に対応する．この実験では $\gamma = 10$, $\theta = 0.03$ を標準の設定とし，そこから値を変動させることで γ 及び θ が各戦略の性能に与える影響を調べた．

また，$\boldsymbol{\Sigma}$ を変えることで $p_q(\boldsymbol{x})$ の等確率線の形状が異なる場合についても評価を行った．各戦略ごとに，$\boldsymbol{q}$ の異なる 100 回の問合せ処理を行い，その平均応答時間を性能の評価基準に用いた．ただし，応答時間は $\boldsymbol{q}$ によって大きく影響を受けるため，100 回分の問合せの $\boldsymbol{q}$ は各戦略ごとにすべて同じものを使用した．事前に作成した U-catalog のエントリ数は，戦略 1 の U-catalog が 609，戦略 2 の U-catalog が 30,888 であった．

今回の実験に用いた問合せ処理プログラムでは，ボロノイ領域や最小包含球の計算などに LEDA 6.1 を使用した．LEDA [15] は，グラフ理論や幾何学計算などの分野における効率的なデータ構造とアルゴリズムを提供する C++のクラスライブラリである．また，数値積分処理には RANDLIB [16] という C 言語の乱数生成ライブラリを用いた．具体的には，RANDLIB により正規分布の確率密度関数に従って大量の乱数を生成し，各乱数がボロノイ領域内に位置しているかどうかを LEDA で提供されている関数を利用して調べた．ボロノイ領域内に位置していた乱数の個数の比率が求める確率の推定値に相当している．この手法は重点サンプリング法 [17] と呼ばれ，モンテカルロ法の一種であるが，通常のモンテカルロ法による計算より高速である．今回の実験では，標準の設定として，1 回の積分計算に対して 1,000,000 個の乱数を発生させて積分値を求めるように設定したが，発生させる乱数の個数を減らした場合についても評価を行った(注11)．戦略 2 で必要となる固有値や行列式の計算には，科学技術計算用の C 言語のライブラリである GNU Scientific Library [18] を用いた．

実験用プログラムの開発には C++を用いた．実験に使用したマシンの CPU は Intel Core 2 Duo E8500 (3.16 GHz)，メモリは 4 GByte，OS は Fedora 11 である．

5.2 実験結果

5.2.1 標準の設定の場合

標準の設定 ($\gamma = 10$，$\theta = 0.03$) における各戦略の応答時間を図 9 に，候補オブジェクト及び解オブジェクトの個数を表 1 に示す．また，ある問合せにおける各戦略の候補オブジェクトを図 10，図 11，図 12 に示す．分布の平均 $\boldsymbol{q}$ は図の中心に位置している．太い線でボロノイ領域が縁取られているオブジェクトが候補オブジェクトであり，ボロノイ領域が黒く塗りつぶされているオブジェクトが解オブジェクトである．図 10，図 12 における方形は，戦略 1 のフィルタリングに用いる θ-領域の包囲方形である．図 9 に示したように，各戦略とも処理時間のほとんどは $\Pr_{NN}(\cdot)$ を求めるための数値積分に費やされていた．これは予想どおりの結果であり，フィルタリングによって数値積分を行うオブジェクトの削減を図るという本手法のアプローチの正しさが確認された．

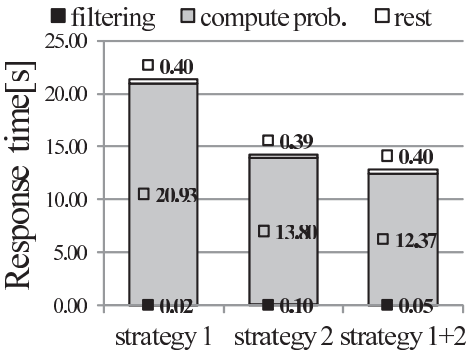

図 9 応答時間 ($\gamma = 10$，$\theta = 0.03$)
Fig. 9 Response time. ($\gamma = 10$, $\theta = 0.03$)

表 1 候補オブジェクト数 ($\gamma = 10$，$\theta = 0.03$)
Table 1 Number of candidates. ($\gamma = 10$, $\theta = 0.03$)

戦略 1	戦略 2	戦略 1+2	解
101.5	64.4	55.7	5.4

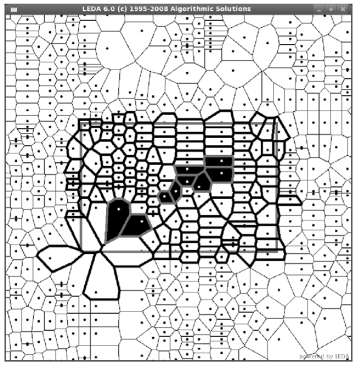

図 10 戦略 1 の候補オブジェクト
Fig. 10 Candidates in Strategy 1.

表 1 より，戦略 1 及び戦略 2 をそれぞれ単独で用いた場合の候補オブジェクト数は，戦略 1 が 101.5 個，戦略 2 が 64.4 個であるが，ハイブリッド戦略では 55.7 個に減っていることが分かる．これは，ハイブリッド戦略では戦略 1 と戦略 2 に共通の候補オブジェクトのみを候補とするため，一方の戦略では候補になるが他

(注11)：参考までに, [5] では主として 10,000 個の乱数を用いているが，実験によっては 1,000,000 個の乱数も用いている．

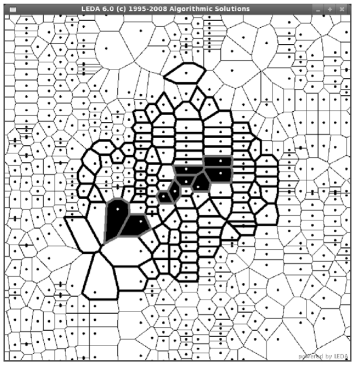

図 11　戦略 2 の候補オブジェクト
Fig. 11　Candidates in Strategy 2.

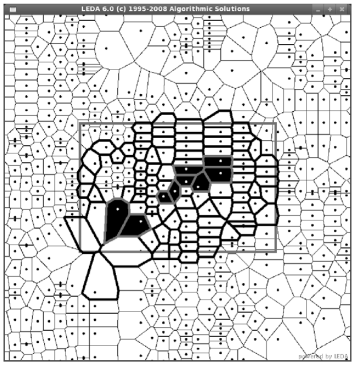

図 12　戦略 1+2 の候補オブジェクト
Fig. 12　Candidates in Strategy 1+2.

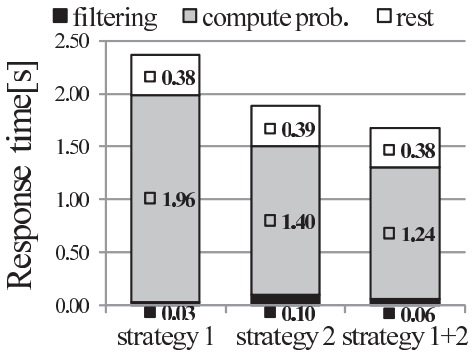

図 13　応答時間（サンプル数：100,000）
Fig. 13　Response time. (No. of samples: 100,000)

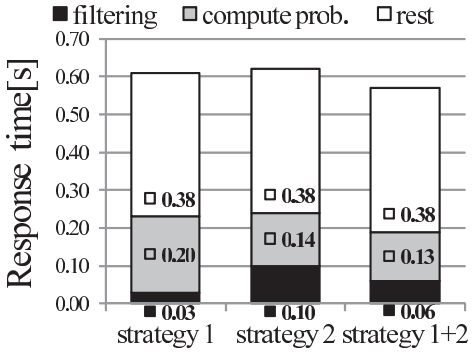

図 14　応答時間（サンプル数：10,000）
Fig. 14　Response time. (No. of samples: 10,000)

方の戦略では棄却されるようなオブジェクトがすべて棄却されるからである．

ハイブリッド戦略では戦略 1 のフィルタリングを行った後に戦略 2 のフィルタリングを行うが，図 9 を見ると，フィルタリングに要した時間は 0.05 秒であり，単純に，戦略 1 でフィルタリングに要した時間 0.02 秒と戦略 2 でフィルタリングに要した時間 0.10 秒の和にはなっていないことが分かる．この理由は，ハイブリッド戦略では戦略 1 のフィルタリングによって残ったオブジェクトに対してのみ戦略 2 のフィルタリングを適用することになるため，戦略 2 を単独で用いる場合よりも戦略 2 のフィルタリングに必要な時間を減らすことができるからである．フィルタリングの適用順を入れ換えた場合，得られる候補オブジェクトの集合は変わらないが，少なくとも戦略 2 を単独で用いる場合の 0.10 秒はフィルタリングに必要となってしまう．そのため，より短時間でフィルタリング処理が可能な「戦略 1 から戦略 2」という順序を採用している．

この実験では 5.4 個という少数の解オブジェクトを返すのにハイブリッド戦略の場合でも約 13 秒かかっており，若干処理時間が長いように思われる．処理時間の短縮に最も効果的な方法は，その大部分を占めている確率の計算に要する時間の短縮であり，数値積分に使用する乱数の個数（以下，サンプル数）を減らせばこれを達成できることは明らかである．つまり，今回の実験ではサンプル数を 1,000,000 個に設定しているが，例えばこれを 100,000 個に減らすことで，計算の精度は悪化するものの確率計算に要する時間をおよそ 1/10 にまで短縮できるはずである．そこで実際に，サンプル数を減少させた場合について実験を行った．その結果を次項で示す．

5. 2. 2　サンプル数を減少させた場合

標準の設定ではサンプル数を 1,000,000 個に設定していたが，これを 100,000 個に設定した場合と，10,000 個に設定した場合の各戦略の応答時間をそれぞれ図 13，図 14 に示す．サンプル数の減少に比例して

確率計算に要する時間が短くなっていることがこれらの図から読み取れる．図 14 のサンプル数 10,000 個の場合には，応答時間がほぼ同等でいずれも 1 秒未満となっている．サンプル数を減らすにつれ，その分計算精度は悪化しており，各戦略ごとに 100 回ずつ，合計 300 回分の問合せ結果の平均誤差は，100,000 個の場合が 0.13，10,000 個の場合が 0.39 であった．ただし，誤差は，標準の設定における解オブジェクトの集合を正解集合とみなし，正解集合には含まれていたが問合せ結果には含まれていなかったオブジェクトの個数と，正解集合には含まれていなかったが問合せ結果には含まれていたオブジェクトの個数の合計値として定義している(注12)．

今回の実験では十分に高い精度を必要としたため，様々な試行に基づいて標準のサンプル数を 1,000,000 個に設定したが，実験結果から，多くの現実的な状況ではサンプル数を減らすことができる可能性が高いと考えている．ただし，サンプル数の設定は，計算時間と計算精度のトレードオフを考慮しながら，適用するアプリケーションの要件やユーザの設定に応じて適切に決定する必要があり，一概には論じられないと考えられる．

サンプル数が多い場合には，確率計算に要する時間が全体の処理時間の大部分を占めるため，各戦略のフィルタリングに要する時間の差はほとんど無視できた．しかしながら，図 14 に示すように，サンプル数がかなり少ない場合には，フィルタリングに要する時間の占める割合が大きくなるため，他の戦略に比べフィルタリングに時間を要する戦略 2 の性能が相対的に悪化することになる．

5.2.3 γ を変動させた場合

γ を変動させると，$p_q(\boldsymbol{x})$ の等確率線の大きさがその形状を保ったまま変化する．等確率線の大きさは問合せオブジェクトの位置のあいまいさの程度を表しており，γ を大きくすることはあいまいさを大きく，逆に γ を小さくすることはあいまいさを小さくすることに対応する．各戦略の応答時間を図 15 に，候補オブジェクト及び解オブジェクトの個数を表 2 に示す．戦略 1 及び戦略 2 ともに，フィルタリングに要する時間はデータ数と U-catalog のエントリ数にのみ依存し，問合せのパラメータには依存しない．また図 9 において「その他」とした時間は，具体的にはデータの読込みに要する時間が大部分であり，こちらも問合せのパラメータには依存しない．先の実験により，サンプル数が極端に少ない場合を除けば，これらの処理時間は

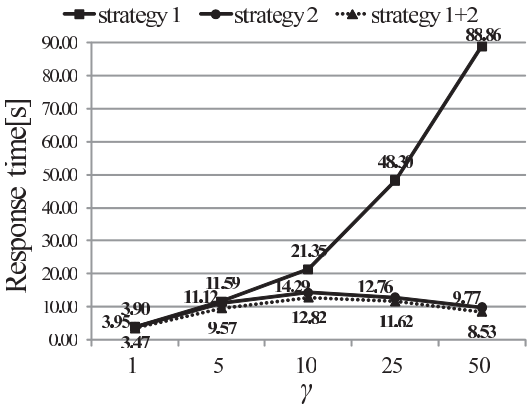

図 15　応答時間（γ を変動）
Fig. 15　Response time. (varying γ)

表 2　候補オブジェクト数（γ を変動）
Table 2　Number of candidates. (varying γ)

	戦略 1	戦略 2	戦略 1+2	解
$\gamma = 1$	15.3	20.7	14.3	4.9
$\gamma = 5$	53.8	50.9	41.7	7.1
$\gamma = 10$	101.5	64.4	55.7	5.4
$\gamma = 25$	232.0	56.4	50.2	3.3
$\gamma = 50$	431.4	42.7	36.4	2.6

確率計算に要する時間に比べ，ごく短時間であることが確認されている．そのため全体の処理時間は，確率計算に要する時間すなわち候補オブジェクト数によってほぼ決まるといえる．ただし，アルゴリズム 2 の 13 行目に示したとおり，戦略 2 ではフィルタリングに用いた最小包含球領域での $p_q^{\top}(\boldsymbol{x})$ の積分値の降順で候補オブジェクトのソートを行うことで解になる可能性が高そうな候補から順に確率を計算できるため，候補オブジェクト数の割に早く処理が終了する場合がある．実際に $\gamma = 1$ の場合について，表 2 を見ると戦略 2 の候補オブジェクト数は戦略 1 に比べ約 35%多くなっているが，図 15 に示した応答時間はほとんど同じである．とはいえ，あくまでもこれは顕著な場合であり，そのほかの場合では候補オブジェクト数にほぼ比例した応答時間になっている．

γ が大きくなると $p_q(\boldsymbol{x})$ の等確率線の大きさも大きくなる．これはすなわち θ-領域が大きくなるということであり，当然ながら戦略 1 では候補オブジェクト数

(注12)：この定義のとおり，本実験における「誤差」は正確な意味での誤差ではない．モンテカルロ法の誤差については，サンプル数を n としたとき，次元によらず $\sqrt{n}$ に比例するオーダであることが知られている [17]．そのため，サンプル数を 1/100 にすると，本来，誤差は 1/10 程度になると推測される．

が増加する．これに対し，表 2 に示されるとおり，戦略 2 における候補オブジェクト数は γ の増大の影響をあまり受けていない．$p_q(\boldsymbol{x})$ の等確率線の大きさが大きくなるということは，それだけなだらかに広がった分布になるということであり，分布の平均 $\boldsymbol{q}$ から比較的近くに位置するオブジェクトについてはボロノイ領域での積分値が減少することになる．つまり，γ の増大によって解でなくなるオブジェクトも存在するため，あいまいさが大きくなったからといって単純に解オブジェクト数が増加するわけではない．実際に，表 2 を見るとそのような結果になっていることが分かる．γ の増大によって解でなくなるかどうかにはボロノイ領域の大きさが関係しており，十分大きければ γ が大きくなっても解であり続け，小さければ解でなくなる可能性が高い．戦略 2 では最小包含球による近似という形でボロノイ領域の大きさという要素を考慮するため，γ の増大によって解でなくなるオブジェクトを棄却できる可能性があるが，戦略 1 では棄却できないため候補オブジェクト数は増加し続ける．図 15 にはその差がはっきりと現れており，γ の増大に伴って戦略 2 の戦略 1 に対する優位性が高くなっているのが分かる．

5.2.4 θ を変動させた場合

各戦略の応答時間を図 16 に，候補オブジェクト及び解オブジェクトの個数を表 3 に示す．θ が大きくなるほど解オブジェクト数が減っているのは，問合せの定義を考えれば当然である．θ が大きくなると，戦略 1 では θ-領域が小さくなるため，戦略 2 では最小包含球領域での積分値が θ を超えるオブジェクトが減るため，ともに候補オブジェクト数は減少する．ただし表 3 を見ると，その減少率は戦略 2 の方が高いことが分かる．そのため図 16 に示されるように，θ の増大に伴って戦略 2 の戦略 1 に対する優位性が高くなっている．

5.2.5 $p_q(\boldsymbol{x})$ の等確率線の形状を変動させた場合

標準の設定では $p_q(\boldsymbol{x})$ の等確率線の形状を長軸と短軸の比が 3 : 1 で傾き 30° の楕円としていたが，この実験では $\boldsymbol{\Sigma}$ を変えることで，等確率線の形状が円の場合と細い楕円（長軸と短軸の比が 9 : 1 で傾き 30° の楕円）の場合について調べた．各戦略の応答時間をそれぞれ図 17，図 18 に，候補オブジェクト及び解オブジェクトの個数をそれぞれ表 4，表 5 に示す．また，

表 3　候補オブジェクト数（θ を変動）
Table 3　Number of candidates. (varying θ)

	戦略 1	戦略 2	戦略 1+2	解
$\theta = 0.01$	137.8	119.7	101.5	18.9
$\theta = 0.02$	115.7	89.6	77.1	9.7
$\theta = 0.03$	101.5	64.4	55.7	5.4
$\theta = 0.04$	93.1	44.6	38.1	3.6
$\theta = 0.05$	85.8	38.9	33.3	2.5

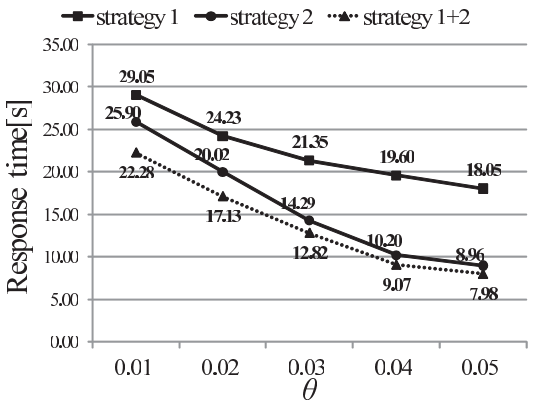

図 16　応答時間（θ を変動）
Fig. 16　Response time. (varying θ)

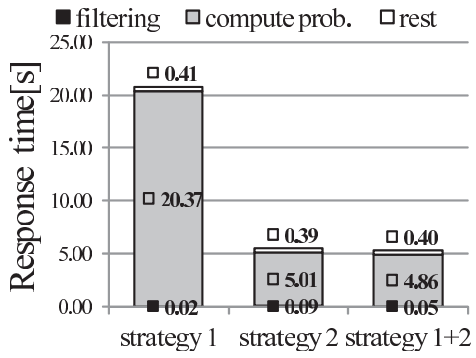

図 17　応答時間（円）
Fig. 17　Response time. (circle)

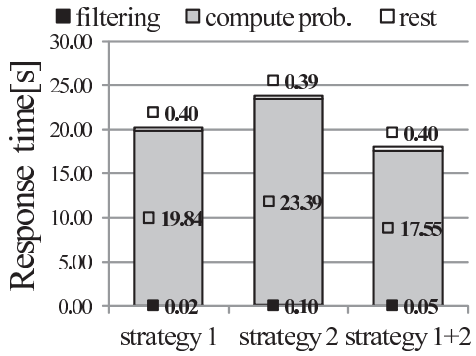

図 18　応答時間（細い楕円）
Fig. 18　Response time. (narrow ellipse)

表 4　候補オブジェクト数（円）
Table 4　Number of candidates. (circle)

戦略 1	戦略 2	戦略 1+2	解
100.6	21.0	20.1	4.1

表 5　候補オブジェクト数（細い楕円）
Table 5　Number of candidates. (narrow ellipse)

戦略 1	戦略 2	戦略 1+2	解
96.8	124.1	84.9	7.5

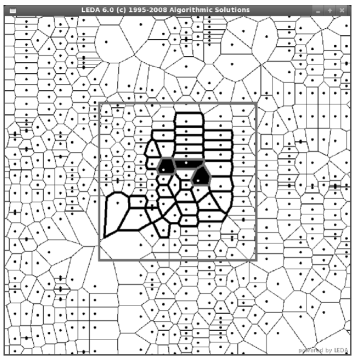

図 19　候補オブジェクト（円）
Fig. 19　Candidates. (circle)

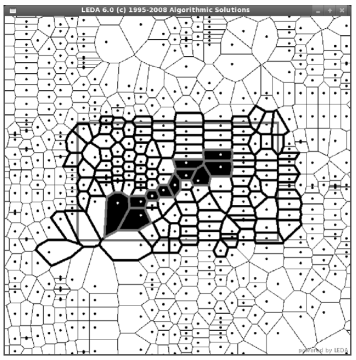

図 20　候補オブジェクト（細い楕円）
Fig. 20　Candidates. (narrow ellipse)

ある問合せにおけるハイブリッド戦略の候補オブジェクトをそれぞれ図 19，図 20 に示す．図 17，図 18 より，等確率線の形状が円である場合には戦略 1 に比べて戦略 2 の方が良い性能を示すが，等確率線の楕円形状を細くした場合には優劣関係が逆転していることが分かる．この理由は以下のとおりである．戦略 2 では最小包含球の領域での積分値を求めるにあたって，$p_q(\boldsymbol{x})$ の代わりにその上限の関数 $p_q^{\top}(\boldsymbol{x})$ の積分値を求めることでフィルタリングを行う．図 8 に示したとおり，同じ確率に対して等確率面を描いたときに，$p_q^{\top}(\boldsymbol{x})$ の等確率面は $p_q(\boldsymbol{x})$ の等確率面に外接する球となる．したがって $p_q(\boldsymbol{x})$ の等確率面の楕円体の形状が球に近い場合には，$p_q^{\top}(\boldsymbol{x})$ の積分値が $p_q(\boldsymbol{x})$ の積分値に近づくため，戦略 2 のフィルタリングの効率が良くなり，逆に楕円体の形状が細い場合には，$p_q^{\top}(\boldsymbol{x})$ の積分値が $p_q(\boldsymbol{x})$ の積分値に比べて大幅に大きくなってしまうため，フィルタリングの効率が悪くなるのである．

6. むすび

本研究では，位置が正規分布によってあいまいな位置情報で表現されているオブジェクトが確定的な位置をもつオブジェクトを対象に最近傍問合せを行うという状況を対象とし，しきい値を導入するなどして通常の最近傍問合せを拡張した確率的最近傍問合せの処理手法を提案した．

本手法では，数値積分によって正確に最近傍オブジェクトとなる確率を求めるまでもなく明らかに確率がしきい値より小さいといえるオブジェクトを求めることで計算コストを削減した．このアプローチに基づく問合せ戦略として，θ-領域に基づく戦略である問合せ戦略 1 と，最小包含球と上限の関数により確率の上限値を求める戦略である問合せ戦略 2 を提案した．

実験では，二つの戦略にそれらのハイブリッド戦略を加えた三つの戦略について，様々なパラメータ設定のもとで比較を行った．その結果，基本的には戦略 2 の方が戦略 1 に比べて性能が良く，特に，サンプル数が多い，問合せオブジェクトの位置のあいまいさが大きい，しきい値が高い，正規分布の等確率面の楕円体の形状が球形に近い，というような状況ではその傾向が顕著であった．ただし，実用性の観点からすると，三つの戦略のうちで最も良い性能を示したハイブリッド戦略によって問合せ処理を行うのがよい．

今後は，移動ロボットなどを用いた現実環境中での実験に取り組みたい．また，問合せ処理のコストを減らすにはモンテカルロ積分のサンプル数を削減することが最も効果的であるが，精度の低下の問題があるため，確率の計算自体の高速化，具体的には正規分布の確率密度関数の積分計算の本質的な高速化についても検討したいと考えている．

謝辞　本研究の一部は，文部科学省科学研究費 (21013023，22300034) の助成による．

文　献

[1] S. Thrun, W. Burgard, and D. Fox, Probabilistic robotics, The MIT Press, 2005.

[2] D. Pfoser and C.S. Jensen, “Capturing the uncertainty of moving-object representations,” Proc. 6th Intl. Symp. on Advances in Spatial Databases (SSD’99), pp.111–131, 1999.
[3] R. Cheng, D.V. Kalashnikov, and S. Prabhakar, “Querying imprecise data in moving object environments,” IEEE Trans. Knowl. Data Eng., vol.16, no.9, pp.1112–1127, 2004.
[4] J. Chen and R. Cheng, “Efficient evaluation of imprecise location-dependent queries,” Proc. ICDE, pp.586–595, 2007.
[5] Y. Tao, X. Xiao, and R. Cheng, “Range search on multidimensional uncertain data,” ACM Trans. Database Syst., vol.32, no.3, 2007.
[6] Y. Ishikawa, Y. Iijima, and J.X. Yu, “Spatial range querying for Gaussian-based imprecise query objects,” Proc. ICDE, pp.676–687, 2009.
[7] H.-P. Kriegel, P. Kunath, and M. Renz, “Probabilistic nearest-neighbor query on uncertain objects,” Proc. DASFAA, pp.337–348, 2007.
[8] R. Cheng, J. Chen, M. Mokbel, and C.-Y. Chow, “Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data,” Proc. ICDE, pp.973–982, 2008.
[9] G. Beskales, M.A. Soliman, and I.F. Ilyas, “Efficient search for the top-k probable nearest neighbors in uncertain databases,” Proc. VLDB, pp.326–339, 2008.
[10] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern classification, 2nd ed., Wiley, 2000.
[11] S.J. Russell and P. Norvig, Artificial intelligence: A modern approach, 2nd ed., Pearson Education, 2003.
[12] M. Kourogi, N. Sakata, T. Okuma, and T. Kurata, “Indoor/outdoor pedestrian navigation with an embedded GPS/RFID/self-contained sensor system,” Proc. ICAT, pp.1310–1321, 2006.
[13] F. Aurenhammer, “Voronoi diagrams — A survey of a fundamental geometric data structure,” ACM Comput. Surv., vol.23, no.3, pp.345–405, 1991.
[14] “TIGER”. http://tiger.census.gov/
[15] “LEDA”. http://www.algorithmic-solutions.com/leda/
[16] “RANDLIB”. http://biostatistics.mdanderson.org/SoftwareDownload/
[17] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, Numerical recipes: The art of scientific computing, 3rd ed., Cambridge University Press, 2007.
[18] “GNU Scientific Library”. http://www.gnu.org/software/gsl/

（平成 21 年 9 月 3 日受付，22 年 1 月 4 日再受付）

飯島　裕一

2008 名大・工・電気電子・情報工学卒．2010 同大大学院情報科学研究科博士前期課程了．現在，アイホン（株）に勤務．

石川　佳治　（正員）

1989 筑波大・第三学群・情報学類卒．1994 同大大学院博士課程工学研究科単位取得退学．同年奈良先端科学技術大学院大学助手．1999 筑波大学電子・情報工学系講師．2004 同助教授．2006 名古屋大学情報連携基盤センター教授．博士（工学）（筑波大学）．データベース，データ工学，情報検索等に興味をもつ．日本データベース学会，情報処理学会，人工知能学会，ACM，IEEE CS 各会員．